Задача A:

Спочатку для зручності знайдемо усі відрізки у вхідному масиві з послідовних однакових чисел. Гра полягає в тому, що ми можемо видаляти деякі відрізки. При цьому у процесі декілька початкових відрізків можуть злитися в один.

Давайте поміркуємо, що ми можемо залишити як відповідь наприкінці гри. З першого абзацу випливає, що це буде один з початкових відрізків, можливо, склеєний з деякими іншими. Давайте знайдемо початковий відрізок з мінімальною сумою і спробуємо залишити як відповідь його. Для цього будемо видаляти перший відрізок вхідної послідовності, доки першим не стане наш відрізок-відповідь. (Зрозуміло, що в процесі такого видалення ніякі два відрізки не зможуть злитися). Після цього робимо такий самий алгоритм з відрізками, що йдуть у кінці масиву. У результаті залишиться один відрізок сума якого мінімальна. Очевидно, меншої суми ми не можемо отримати, адже нам не можна розрізати початкові відрізки.

Отже, відповіддю буде мінімальна сума відрізка з однакових чисел для вхідної послідовності.

Задача B:

Давайте напишемо таку нескладну динаміку:

$$dp[i][j]\ 1 <= i <= n;\ 0 <= j <= k$$

де ***dp[i][j]*** буде зберігати максимальну потужність системи з ***i*** телескопів, якщо останній телескоп ми взяли з позиції ***j***. Перераховувати динаміку не важко -- просто переберемо телескоп ***p***, який був передостаннім, та напишемо перехід:

$$dp[i][j] = min(dp[i][j], dp[p][j - 1] + (a[i]\ xor\ a[p]))$$

Відповіддю буде максимум серед усіх $dp[i][k]$.

Ця динаміка має складність ***O(n * n * k)***, що не дозволяє нам розв’язати задачу для обмежень більших за 300. Проте можна помітити, що для останнього блоку нам не потрібно видаляти телескопи, адже видалення телескопу ніяк не може збільшити загальну потужність. Отже, для останнього блоку достатньо порахувати функцію для вхідного масиву та вивести результат.

Задача С:

Спочатку навчимося знаходити відповідь (оптимальне розбиття на групи) для всього масиву. Якщо масив складається з одного числа, то відповідь очевидна. Інакше справедливе наступне твердження:

Нехай mxA -- максимальний елемент множини А,

$\{A / mxA\}$ -- множина А без її максимального елементу,

$bestAnswer(A)$ -- найкраща відповідь для множини А.

Тоді

$$bestAnswer(A) = \frac{mxA + bestAnswer(\{A / mxA\})}{2}$$

Тобто ми ділимо множину на дві частини: максимум та всі інші елементи, причому всі інші елементи рекурсивно розбиваємо оптимально.

Очевидно, що така формула дає кращий результат, ніж середнє арифметичне для множини.

Складніше довести, що неможливо знайти краще розбиття.

Інтуїтивно можна зрозуміти, що формула розбиття схожа на формулу середнього арифметичного множини, але якщо в середньому арифметичному кожен елемент входить з коефіцієнтом $1/n$, то розбиття дає нам можливість ці коефіцієнти змінити -- збільшити для одних чисел та зменшити для інших.

Якщо рекурсивно розвернути формулу найкращої відповіді, отримаємо:

$$bestAnswer(A) = mxA / 2 + mxA_2 / 4 + mxA_3 / 8 + ... + mxA_{n-1} / 2^{n-1} + mxA_n / 2^{n-1}$$

де mxA_x – елемент множини А з номером *x*, якщо відсортувати множину за незростанням (*x*-тий максимум множини).

Ми зробили коефіцієнт біля максимальних елементів масиву якомога більшим, а біля мінімальних -- якомога меншим. Можливо, це і буде оптимальною відповіддю?

(Так, буде. Цього припущення вистачило автору розбору для розв'язання даної задачі. Коректність оптимальної формули можна довести строго математично, уважно розглянувши всі випадки та переконавшись, що не можна покращити розбиття. Вправа із зірочкою для кмітливих читачів.)

Тоді, щоб порахувати відповідь для масиву, можна скористатися розгорнутою формулою вище. Як навчитися рахувати відповідь для відрізку?

Необхідно помітити, що нас просять знайти відповідь з точністю до 10^{-4}. А, наприклад, уже п'ятидесятий елемент розгорнутої формули буде не більший за $10^6 / 10^{2^{50}}$ ~ 10^{-9}, і ця оцінка буде стрімко (експоненційно) падати для подальших елементів, тому вони абсолютно не впливають на відповідь для заданої точності.

Отже, для розв'язку задачі на відрізку необхідно швидко знаходити 50 максимальних чисел на відрізку та застосувати обрізану розгорнуту формулу. Це можна ефективно реалізувати Деревом Відрізків зі зберіганням максимуму та його позиції на відрізку.